UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/526,750	11/14/2005	Thorsten Pferdekaemper	07781.0220	6924

60668 7590 03/18/2009

SAP / FINNEGAN, HENDERSON LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

EXAMINER	
SANDERS, AARON J	

ART UNIT	PAPER NUMBER
2168	

MAIL DATE	DELIVERY MODE
03/18/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/526,750	PFERDEKAEMPER ET AL.	
	Examiner	Art Unit	
	AARON SANDERS	2168	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *05 January 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-9,11,12 and 15-31* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-9,11,12 and 15-31* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *07 March 2005* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *01/05/2009 and 03/11/2009*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 5 January 2009 has been entered.

Response to Amendment

The amendment filed 2 December 2008 has been entered. Claims 1-9, 11-12, and 15-31 are pending. Claims 1, 11-12, and 23 are currently amended. Claims 10 and 13-14 are cancelled. No claims are new.

Double Patenting

Claims 1, 3-4, 11-12, 16-17, 23, and 25-26 of this application conflict with claims 1-6, 8-13, 15-20, and 22-25 of Application No. 10/656,208. 37 CFR 1.78(b) provides that when two or more applications filed by the same applicant contain conflicting claims, elimination of such claims from all but one application may be required in the absence of good and sufficient reason for their retention during pendency in more than one application. Applicant is required to either cancel the conflicting claims from all but one application or maintain a clear line of demarcation between the applications. See MPEP § 822.

The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

Claims 1, 3-4, 11-12, 16-17, 23, and 25-26 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over

claims 1-6, 8-13, 15-20, and 22-25 of copending Application 10/656,208. Although the conflicting claims are not identical, they are not patentably distinct from each other because independent claims 1, 8, 15, and 22 of 10/656,208 are broader (i.e. fewer limitations of the same invention) than independent claims 1, 11, 12, and 23 of 10/526,750. While some of the claim terms are not the same (e.g. "checking" instead of "determining" and "granting access" instead of "performing the read and/or write access"), the functionality is the same. Thus, it would have been obvious to one of ordinary skill in the art that the fewer limitations and slightly different language of 10/656,208 would read on the method, system, and computer readable storage medium of 10/526,750.

Claims Comparison Table	
10/526,750	10/656,208
1	1, 2, 6
3	3, 4
4	5
11	8, 9, 13
12	15, 16, 20
16	17, 18
17	19
23	22
25	10, 11, 23
26	12, 24, 25

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

Claim Objections

As per claims 1, 11-12, and 23, the phrase "read and/or write" is improper because it is unclear whether the limitation permits both read and write access or only one or the other.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-9, 11-12, and 15-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lenz, U.S. 5,566,319, in view of Lakhamraju et al., U.S. 6,343,296.

1. Lenz teaches "*A method comprising*," see col. 1, l. 66 – col. 2, l. 23, "a method of controlling access to data in storage which data and storage are shared by a plurality of processors."

Lenz teaches "*storing an identifier (ID) in a second lock object, the ID being associated with a data object stored in a first storage location in a computer system*," see Fig. 3, where the claimed "second lock object" is the referenced array of shared data records "006," "007," and "008," the claimed "ID" is the referenced "007," the claimed

"data object" is the referenced "account record 007," and the claimed "first storage" is the referenced shared storage 3-9.

Lenz teaches "*determining whether the ID is contained in a first lock object*," see Fig. 3 and col. 3, ll. 1-10, "By using the search key '007' during the execution of the lock instruction, the part of the lock file containing the control field for record 007 is addressed for writing," where the claimed "first lock object" is the referenced "Lock-File" 3-1.

Lenz teaches "*if the ID is contained in the first lock object*," see Fig. 3 and col. 3, ll. 1-10, "By using the search key '007' during the execution of the lock instruction, the part of the lock file containing the control field for record 007 is addressed for writing." Lenz does not teach "*determining whether a link to a second storage location having a copy of the data object is assigned to the ID in the first lock object*." Lakhamraju does, however, see col. 2, ll. 9-16, "However, the data record itself is not found merely from the logical reference. Rather a mapping procedure is executed to return the physical location of an object. It is only the one cross reference in the map that requires updating when an object is moved from one physical location to another," where the claimed "link" is the referenced "logical reference," the claimed "second storage location" is the referenced "another" "physical location," and the claimed "first lock object" is the referenced "map." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Lakhamraju's teachings would have allowed Lenz's method to gain a faster means of updating references to a relocated object, see col. 2, ll. 17-29.

Lenz teaches "*and if the ID is not contained in the first lock object, performing the read and/or write access on the data object*," see Fig. 3 and col. 3, ll. 51-62, "If the examination of the status identification code SKC shows that the write operation into the control field for account record 007, refer to 3-2 for instance, may be executed, this write operation is carried out and the status identification code SKC is updated according to the write request."

Lenz does not explicitly teach "*and if the link is assigned to the ID.*" Lakhamraju does, however, see col. 2, ll. 9-16, "However, the data record itself is not found merely from the logical reference. Rather a mapping procedure is executed to return the physical location of an object. It is only the one cross reference in the map that requires updating when an object is moved from one physical location to another." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Lakhamraju's teachings would have allowed Lenz's method to gain a faster means of updating references to a relocated object, see col. 2, ll. 17-29. Lenz does teach "*skipping performing a read and/or write access on the data object.*" see Fig. 3 and col. 5, ll. 51-55, "SKC could be 1: block not empty (at least one control field not empty), i.e. an access right may only be granted after detailed examination," where the referenced "access" may not be granted after the "detailed examination."

Lenz does not explicitly teach "*and if the link is not assigned to the ID.*" Lakhamraju does, however, see col. 2, ll. 9-16, "However, the data record itself is not found merely from the logical reference. Rather a mapping procedure is executed to return the physical location of an object. It is only the one cross reference in the map that

requires updating when an object is moved from one physical location to another." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Lakhamraju's teachings would have allowed Lenz's method to gain a faster means of updating references to a relocated object, see col. 2, ll. 17-29. Lenz does teach "*deleting the ID from the first lock object*," see e.g. Fig. 4 and col. 4, l. 34 – col. 5, l. 3, "the status identification code is updated accordingly in step 4-10, for example, to the effect that the control field is now empty and may be overwritten without having to be read later on." Lenz further teaches "*and performing the read and/or write access on the data object*," see Fig. 3 and col. 5, ll. 49-50, "SKC could be 0: block empty (all control fields empty), i.e. an access right may be granted immediately."

2. Lenz teaches "*The method of claim 1, wherein the first lock object is a file stored on a nonvolatile storage means*," see Fig. 3 and col. 1, l. 66 – col. 2, l. 23, "Each of the processors has a local main storage. The shared storage is located outside of the main storage and stores a lock file."

3. Lenz teaches "*The method of claim 1, wherein the first lock object comprises a table having a first column for the ID and a second column for the link of the ID to the second storage location*," see Fig. 3, "Lock-File" 3-1.

4. Lenz teaches "*The method of claim 1, wherein each data object comprises one or more fields of one or more tables and wherein the ID comprises one or more key fields of the one or more tables*," see Fig. 5.

5. Lenz teaches "*The method of claim 4, wherein the link is a filename or a link to a file*," see Fig. 3 and col. 1, l. 66 – col. 2, l. 23, "Each of the control fields is associated with a corresponding data address."

6. Lenz teaches "*The method of claim 1, wherein the first lock object is created by a data moving process*," see col. 1, l. 66 – col. 2, l. 23, "a method of controlling access to data in storage which data and storage are shared by a plurality of processors."

7. Lenz teaches "*The method of claim 1, wherein the second lock object is stored in a volatile storage means*," see Fig. 3 and col. 1, l. 66 – col. 2, l. 23, "Each of the processors has a local main storage. The shared storage is located outside of the main storage and stores a lock file."

8. Lenz teaches "*The method of claim 1, wherein the second lock object is a data array*," see Fig. 3, the array of "Shared Data Record[s]" for "006," "007," "008," etc.

9. Lenz teaches "*The method of claim 8 wherein the data array is one dimensional*," see Fig. 3, the array of "Shared Data Record[s]" for "006," "007," "008," etc.

11. Lenz teaches "*A computer system for processing data, comprising: memory means for storing program instructions*," see Fig. 3, "Local Main Storage 1" 3-6.

Lenz teaches "*input means for entering data*," see Fig. 3, "Writing."

Lenz teaches "*storage means for storing data*," see Fig. 3, shared storage 3-9.

Lenz teaches "*a processor responsive to the program instructions, wherein the program instructions comprise program code means for performing a method for accessing a data object having an identifier (ID) and stored in a first storage location*,"

see Fig. 3 and col. 1, l. 66 – col. 2, l. 23, "a method of controlling access to data in storage which data and storage are shared by a plurality of processors."

Lenz teaches "*the method comprising: storing the ID in a second lock object*," see Fig. 3, where the claimed "second lock object" is the referenced array of shared data records "006," "007," and "008" and the claimed "ID" is the referenced "007."

Lenz teaches "*determining whether the ID is contained in a first lock object*," see Fig. 3 and col. 3, ll. 1-10, "By using the search key '007' during the execution of the lock instruction, the part of the lock file containing the control field for record 007 is addressed for writing," where the claimed "first lock object" is the referenced "Lock-File" 3-1.

Lenz teaches "*if the ID is contained in the first lock object*," see Fig. 3 and col. 3, ll. 1-10, "By using the search key '007' during the execution of the lock instruction, the part of the lock file containing the control field for record 007 is addressed for writing." Lenz does not teach "*determining whether a link to a second storage location having a copy of the data object is assigned to the ID in the first lock object*." Lakhamraju does, however, see col. 2, ll. 9-16, "However, the data record itself is not found merely from the logical reference. Rather a mapping procedure is executed to return the physical location of an object. It is only the one cross reference in the map that requires updating when an object is moved from one physical location to another," where the claimed "link" is the referenced "logical reference," the claimed "second storage location" is the referenced "another" "physical location," and the claimed "first lock object" is the referenced "map." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references

because Lakhamraju's teachings would have allowed Lenz's method to gain a faster means of updating references to a relocated object, see col. 2, ll. 17-29.

Lenz teaches "*and if the ID is not contained in the first lock object, performing the read and/or write access on the data object*," see Fig. 3 and col. 3, ll. 51-62, "If the examination of the status identification code SKC shows that the write operation into the control field for account record 007, refer to 3-2 for instance, may be executed, this write operation is carried out and the status identification code SKC is updated according to the write request."

Lenz does not explicitly teach "*and if the link is assigned to the ID*." Lakhamraju does, however, see col. 2, ll. 9-16, "However, the data record itself is not found merely from the logical reference. Rather a mapping procedure is executed to return the physical location of an object. It is only the one cross reference in the map that requires updating when an object is moved from one physical location to another." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Lakhamraju's teachings would have allowed Lenz's method to gain a faster means of updating references to a relocated object, see col. 2, ll. 17-29. Lenz does teach "*skipping performing a read and/or write access on the data object*." see Fig. 3 and col. 5, ll. 51-55, "SKC could be 1: block not empty (at least one control field not empty), i.e. an access right may only be granted after detailed examination," where the referenced "access" may not be granted after the "detailed examination."

Lenz does not explicitly teach "*and if the link is not assigned to the ID*." Lakhamraju does, however, see col. 2, ll. 9-16, "However, the data record itself is not

found merely from the logical reference. Rather a mapping procedure is executed to return the physical location of an object. It is only the one cross reference in the map that requires updating when an object is moved from one physical location to another." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Lakhamraju's teachings would have allowed Lenz's method to gain a faster means of updating references to a relocated object, see col. 2, ll. 17-29. Lenz does teach "*deleting the ID from the first lock object*," see e.g. Fig. 4 and col. 4, l. 34 – col. 5, l. 3, "the status identification code is updated accordingly in step 4-10, for example, to the effect that the control field is now empty and may be overwritten without having to be read later on." Lenz further teaches "*and performing the read and/or write access on the data object*," see Fig. 3 and col. 5, ll. 49-50, "SKC could be 0: block empty (all control fields empty), i.e. an access right may be granted immediately."

12. Lenz teaches "*A computer readable storage medium comprising instructions for performing a method comprising*," see col. 1, l. 66 – col. 2, l. 23, "a method of controlling access to data in storage which data and storage are shared by a plurality of processors."

Lenz teaches "*storing an identifier (ID) in a second lock object, the ID being associated with a data object stored in a first storage location in a computer system*," see Fig. 3, where the claimed "second lock object" is the referenced array of shared data records "006," "007," and "008," the claimed "ID" is the referenced "007," the claimed "data object" is the referenced "account record 007," and the claimed "first storage" is the referenced shared storage 3-9.

Lenz teaches "*determining whether the ID is contained in a first lock object*," see Fig. 3 and col. 3, ll. 1-10, "By using the search key '007' during the execution of the lock instruction, the part of the lock file containing the control field for record 007 is addressed for writing," where the claimed "first lock object" is the referenced "Lock-File" 3-1.

Lenz teaches "*if the ID is contained in the first lock object*," see Fig. 3 and col. 3, ll. 1-10, "By using the search key '007' during the execution of the lock instruction, the part of the lock file containing the control field for record 007 is addressed for writing." Lenz does not teach "*determining whether a link to a second storage location having a copy of the data object is assigned to the ID in the first lock object.*" Lakhamraju does, however, see col. 2, ll. 9-16, "However, the data record itself is not found merely from the logical reference. Rather a mapping procedure is executed to return the physical location of an object. It is only the one cross reference in the map that requires updating when an object is moved from one physical location to another," where the claimed "link" is the referenced "logical reference," the claimed "second storage location" is the referenced "another" "physical location," and the claimed "first lock object" is the referenced "map." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Lakhamraju's teachings would have allowed Lenz's method to gain a faster means of updating references to a relocated object, see col. 2, ll. 17-29.

Lenz teaches "*and if the ID is not contained in the first lock object, performing the read and/or write access on the data object*," see Fig. 3 and col. 3, ll. 51-62, "If the examination of the status identification code SKC shows that the write operation into the

control field for account record 007, refer to 3-2 for instance, may be executed, this write operation is carried out and the status identification code SKC is updated according to the write request."

Lenz does not explicitly teach "*and if the link is assigned to the ID.*" Lakhamraju does, however, see col. 2, ll. 9-16, "However, the data record itself is not found merely from the logical reference. Rather a mapping procedure is executed to return the physical location of an object. It is only the one cross reference in the map that requires updating when an object is moved from one physical location to another." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Lakhamraju's teachings would have allowed Lenz's method to gain a faster means of updating references to a relocated object, see col. 2, ll. 17-29. Lenz does teach "*skipping performing a read and/or write access on the data object.*" see Fig. 3 and col. 5, ll. 51-55, "SKC could be 1: block not empty (at least one control field not empty), i.e. an access right may only be granted after detailed examination," where the referenced "access" may not be granted after the "detailed examination."

Lenz does not explicitly teach "*and if the link is not assigned to the ID.*" Lakhamraju does, however, see col. 2, ll. 9-16, "However, the data record itself is not found merely from the logical reference. Rather a mapping procedure is executed to return the physical location of an object. It is only the one cross reference in the map that requires updating when an object is moved from one physical location to another." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Lakhamraju's

teachings would have allowed Lenz's method to gain a faster means of updating references to a relocated object, see col. 2, ll. 17-29. Lenz does teach "*deleting the ID from the first lock object*," see e.g. Fig. 4 and col. 4, l. 34 – col. 5, l. 3, "the status identification code is updated accordingly in step 4-10, for example, to the effect that the control field is now empty and may be overwritten without having to be read later on." Lenz further teaches "*and performing the read and/or write access on the data object*," see Fig. 3 and col. 5, ll. 49-50, "SKC could be 0: block empty (all control fields empty), i.e. an access right may be granted immediately."

15. Lenz teaches "*The computer readable storage medium of claim 12, wherein the first lock object is a file stored on a nonvolatile storage means*," see Fig. 3 and col. 1, l. 66 – col. 2, l. 23, "Each of the processors has a local main storage. The shared storage is located outside of the main storage and stores a lock file."

16. Lenz teaches "*The computer readable storage medium of claim 12, wherein the first lock object comprises a table having a first column for the ID and a second column for the link of the ID to the second storage location*," see Fig. 3, "Lock-File" 3-1.

17. Lenz teaches "*The computer readable storage medium of claim 12, wherein each data object comprises one or more fields of one or more tables and wherein the ID comprises one or more key fields of the one or more tables*," see Fig. 5.

18. Lenz teaches "*The computer readable storage medium of claim 12, wherein the link is a filename or a link to a file*," see Fig. 3 and col. 1, l. 66 – col. 2, l. 23, "Each of the control fields is associated with a corresponding data address."

19. Lenz teaches "*The computer readable storage medium of claim 12, wherein the first lock object is created by a data moving process*," see col. 1, l. 66 – col. 2, l. 23,

"a method of controlling access to data in storage which data and storage are shared by a plurality of processors."

20. Lenz teaches "*The computer readable storage medium of claim 12, wherein the second lock object is stored in a volatile storage means*," see Fig. 3 and col. 1, l. 66 – col. 2, l. 23, "Each of the processors has a local main storage. The shared storage is located outside of the main storage and stores a lock file."

21. Lenz teaches "*The computer readable storage medium of claim 12, wherein the second lock object is a data array*," see Fig. 3, the array of "Shared Data Record[s]" for "006," "007," "008," etc.

22. Lenz teaches "*The computer readable storage medium of claim 21, wherein the data array is one dimensional*," see Fig. 3, the array of "Shared Data Record[s]" for "006," "007," "008," etc.

23. Lenz teaches "*A computer system comprising*," see Fig. 3 and col. 1, l. 66 – col. 2, l. 23, "a method of controlling access to data in storage which data and storage are shared by a plurality of processors."

Lenz teaches "*a processor responsive to program instructions*," see Fig. 3 and col. 1, l. 66 – col. 2, l. 23, "a method of controlling access to data in storage which data and storage are shared by a plurality of processors."

Lenz teaches "*means for storing an identifier (ID) in a second lock object, the ID being associated with a data object stored in a first storage location in a computer system*," see Fig. 3, where the claimed "second lock object" is the referenced array of shared data records "006," "007," and "008," the claimed "ID" is the referenced "007,"

the claimed "data object" is the referenced "account record 007," and the claimed "first storage" is the referenced shared storage 3-9.

Lenz teaches "*means for determining whether the ID is contained in a first lock object*," see Fig. 3 and col. 3, ll. 1-10, "By using the search key '007' during the execution of the lock instruction, the part of the lock file containing the control field for record 007 is addressed for writing," where the claimed "first lock object" is the referenced "Lock-File" 3-1.

Lenz teaches "*if the ID is contained in the first lock object*," see Fig. 3 and col. 3, ll. 1-10, "By using the search key '007' during the execution of the lock instruction, the part of the lock file containing the control field for record 007 is addressed for writing." Lenz does not teach "*determining whether a link to a second storage location having a copy of the data object is assigned to the ID in the first lock object*." Lakhamraju does, however, see col. 2, ll. 9-16, "However, the data record itself is not found merely from the logical reference. Rather a mapping procedure is executed to return the physical location of an object. It is only the one cross reference in the map that requires updating when an object is moved from one physical location to another," where the claimed "link" is the referenced "logical reference," the claimed "second storage location" is the referenced "another" "physical location," and the claimed "first lock object" is the referenced "map." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Lakhamraju's teachings would have allowed Lenz's method to gain a faster means of updating references to a relocated object, see col. 2, ll. 17-29.

Lenz teaches "*and if the ID is not contained in the first lock object, performing the read and/or write access on the data object*," see Fig. 3 and col. 3, ll. 51-62, "If the examination of the status identification code SKC shows that the write operation into the control field for account record 007, refer to 3-2 for instance, may be executed, this write operation is carried out and the status identification code SKC is updated according to the write request."

Lenz does not explicitly teach "*and if the link is assigned to the ID.*" Lakhamraju does, however, see col. 2, ll. 9-16, "However, the data record itself is not found merely from the logical reference. Rather a mapping procedure is executed to return the physical location of an object. It is only the one cross reference in the map that requires updating when an object is moved from one physical location to another." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Lakhamraju's teachings would have allowed Lenz's method to gain a faster means of updating references to a relocated object, see col. 2, ll. 17-29. Lenz does teach "*skipping performing a read and/or write access on the data object.*" see Fig. 3 and col. 5, ll. 51-55, "SKC could be 1: block not empty (at least one control field not empty), i.e. an access right may only be granted after detailed examination," where the referenced "access" may not be granted after the "detailed examination."

Lenz does not explicitly teach "*and if the link is not assigned to the ID.*" Lakhamraju does, however, see col. 2, ll. 9-16, "However, the data record itself is not found merely from the logical reference. Rather a mapping procedure is executed to return the physical location of an object. It is only the one cross reference in the map that

requires updating when an object is moved from one physical location to another." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Lakhamraju's teachings would have allowed Lenz's method to gain a faster means of updating references to a relocated object, see col. 2, ll. 17-29. Lenz does teach "*deleting the ID from the first lock object*," see e.g. Fig. 4 and col. 4, l. 34 – col. 5, l. 3, "the status identification code is updated accordingly in step 4-10, for example, to the effect that the control field is now empty and may be overwritten without having to be read later on." Lenz further teaches "*and performing the read and/or write access on the data object*," see Fig. 3 and col. 5, ll. 49-50, "SKC could be 0: block empty (all control fields empty), i.e. an access right may be granted immediately."

24. Lenz teaches "*The computer system of claim 23, wherein first lock object is a file stored on a nonvolatile storage means*," see Fig. 3 and col. 1, l. 66 – col. 2, l. 23, "Each of the processors has a local main storage. The shared storage is located outside of the main storage and stores a lock file."

25. Lenz teaches "*The computer system of claim 23, wherein the first lock object comprises a table having a first column for the ID and a second column for the link of the ID to the second storage location*," see Fig. 3, "Lock-File" 3-1.

26. Lenz teaches "*The computer system of claim 23, wherein each data object comprises one or more fields of one or more tables and wherein the ID comprises one or more key fields of the one or more tables*," see Fig. 5.

27. Lenz teaches "*The computer system of claim 23, wherein the link is a filename or a link to a file*," see Fig. 3 and col. 1, l. 66 – col. 2, l. 23, "Each of the control fields is associated with a corresponding data address."

28. Lenz teaches "*The computer system of claim 23, wherein the first lock object is created by a data moving process*," see col. 1, l. 66 – col. 2, l. 23, "a method of controlling access to data in storage which data and storage are shared by a plurality of processors."

29. Lenz teaches "*The computer system of claim 23, wherein the second lock object is stored in a volatile storage means*," see Fig. 3 and col. 1, l. 66 – col. 2, l. 23, "Each of the processors has a local main storage. The shared storage is located outside of the main storage and stores a lock file."

30. Lenz teaches "*The computer system of claim 23, wherein the second lock object is a data array*," see Fig. 3, the array of "Shared Data Record[s]" for "006," "007," "008," etc.

31. Lenz teaches "*The computer system of claim 30, wherein the data array is one dimensional*," see Fig. 3, the array of "Shared Data Record[s]" for "006," "007," "008."

Response to Arguments

As per Applicant's argument that Lenz does not teach "*determining whether the ID is contained in a first lock object*," the Examiner respectfully disagrees. Specifically, the Examiner cited Fig. 3 and col. 3, ll. 1-10, "By using the search key '007' during the execution of the lock instruction, the part of the lock file containing the control field for record 007 is addressed for writing," where the claimed "first lock object" is the

referenced lock file 3-1. Lenz's method uses the identifier (ID) 007 as a "search key" to find the "part of the lock file containing the control field for record 007." Searching the lock file requires comparing the search key 007 to the ID fields of the lock file and determining which, if any, contains "007." Thus, Lenz teaches "determining whether the ID is contained in a first lock object."

As per Applicant's argument that Lakhamraju does not teach "*determining whether a link to a second storage location having a copy of the data object is assigned to the ID in the first lock object*," the Examiner respectfully disagrees. Specifically, the Examiner cited col. 2, ll. 9-16, "However, the data record itself is not found merely from the logical reference. Rather a mapping procedure is executed to return the physical location of an object. It is only the one cross reference in the map that requires updating when an object is moved from one physical location to another," where the claimed "link" is the referenced "logical reference," the claimed "second storage location" is the referenced "another" "physical location," and the claimed "first lock object" is the referenced "map." The logical reference to a migrated object stored in the map will point to that object's new location. When an application references an object, it must first determine whether a logical reference to that object exists in the map. Thus, Lakhamraju teaches "determining whether a link to a second storage location having a copy of the data object is assigned to the ID in the first lock object."

Applicant should note that "Language that suggests or makes optional but does not require steps to be performed or does not limit a claim to a particular structure does not limit the scope of a claim or claim limitation," see MPEP 2106(II)(C). Here, one branch of an "if" statement is optional.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aaron Sanders whose telephone number is 571-270-1016. The examiner can normally be reached on M-F 9:00a-4:00p.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim Vo can be reached on 571-272-3642. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Tim T. Vo/
Supervisory Patent Examiner, Art Unit 2168

/Aaron Sanders/
Examiner, Art Unit 2168
10 March 2009